REMARKS

The Office action was issued before the Federal Circuit's decision in the case of *In re Bilski*, 88 USPQ 2d 1385 (Fed. Cir. 2008). Accordingly, in view of the Office's new policies regarding patent eligibility under 35 USC §101, the Examiner will note that method claim 1 has been amended, proactively, to recite that the method steps are "machine-implemented." This amendment is made solely to address any potential *Bilski*-related issue.

Claims 19-28 are rejected under 35 U.S.C. §103(a) as being unpatentable over Melchione et al, U.S. Publication No. 2004/0006586, further in view of Ulrich et al, U.S. Patent No. 7,054,927. This rejection is traversed.

A determination regarding alleged obviousness under 35 USC §103(a) requires an analysis of the "scope and content" of the cited art.

Melchione describes a system for distributing software components to nodes distributed throughout a network. As illustrated in Figure 2, an application service provider provides services for administrating instances of the software 212 via a data center 232. A particular instance of the software might be a software release that needs to be delivered to one or more of the nodes in the network. Each node includes a computer 224 having an agent 228 that communicates with the data center 232 to assist in the process. According to paragraph [0070], the distribution of the software 212 can be accomplished in several ways. In one approach, an administrator specifies which nodes are to receive which releases. Automated processing can then be used to distribute the software to those nodes, and the source copy of the software (that is to be distributed) may reside anywhere in the network. Thus, if desired, a node can be designated to provide the software to another "peer" node within the network, thus alleviating the burden of distributing the software releases from a central location (which could end up as a bottleneck). This particular operation is described in paragraph [0135], which indicates that a "central database can be employed to enable agents to obtain software from other agents."

Ulrich describes a dynamically distributed file system. Among other features, the system proactively positions objects for initial load balancing and then continuously to position objects in a manner to accomplish active load balancing for the existing objects throughout the system. The load balancing can be "capacity balancing, throughput balancing, or both." (See, column 4,

line 44, through column 5, line 2). The system comprises servers or clusters, together with a Distributed File Storage System (DFSS) that is distributed over such physical resources.

In operation, the system provides "adaptive load balancing" by predicting future server workload and reallocating distributed server resources in response to the predicted workload. The predicted "future workload" can be based on server workload, file access statistics, server utilization statistics, network utilization statistics, and the like, and those predictions are used to proactively manage resources to improve performance and capacity usage. Software resident on each server collects statistics regarding file accesses and server resource utilization. This includes information regarding access frequency, access bandwidth and access locality for the individual files, the loading of each disk controller and disk storage element in terms of CPU utilization, data transfer bandwidth, transactions per second, and the loading of each network element in terms of network latency and data transfer bandwidth. The collected statistics are filtered to generate the prediction of future file and resource utilization (i.e. workload). (See column 15, line 39 through column 16, line 12).

According to Ulrich, the "predicted workload is then used to develop a plan [ ] where to move content (files) between storage elements and where to direct client accesses to controllers in such a manner that overall workload is distributed as evenly as possible, resulting in best overall load balance and distributed server performance." (See, column 16, lines 19-24).

Turning to the "differences" between the cited art and the claims, the Examiner admits, correctly, that Melchione does <u>not</u> disclose the following limitations of claim 19:

> "identifying values for (i) a flit-capacity, and (ii) a memory capacity, where a flit is an arbitrary unit of work representing resource usage on the CDN server;
>
> using the values to generate a weighted mapping of web applications to manager processes for the set of CDN servers such that the flit and memory capacities for each CDN server are not exceeded; and
>
> servicing requests at the CDN servers in proportion to the weighted mapping."

According to the Examiner, however, Ulrich teaches these remaining limitations. This conclusion, however, is incorrect.

The Examiner's attention is directed to the clause "a weighted mapping of web applications to manager processes" in particular. *This subject matter is not disclosed or suggested in Ulrich.* In that system, as described above, "content (files)" are moved between storage elements and "client accesses to controllers" are distributed so that, taken together, such actions provide the desired load balancing. According to claim 19, however, the resource-related values (for "flit-capacity" and "memory capacity") are used to generate a "weighted mapping" of "*web applications*" to "*manager processes*" where, according to the claim preamble, "each CDN server" is provisioned with "a manager process together with an application server on which one or more web applications are capable of being loaded and executed." In other words, according to claim 19, there are CDN servers that each support a manager process and one or more web applications; *the weighted mapping then allocates the "web applications" to the "manager processes."* Ulrich does not disclose or suggest any such operation.

Of course, one cannot show non-obviousness by attacking references individually where the rejection is based on a combination of the references; *In re Keller*, 642 F.2d 413, 416 (CCPA 1981). Applicants are not attacking the rejection on this basis.

Rather, a test for obviousness is what the combined teachings of the references would have suggested to those of ordinary skill in the art, *Id.* at 426. Here, the "combined teachings" of Melchoine/Ulrich describe a system for distributing software components or other content files to nodes distributed throughout a distributed network together with an adaptive load balancing scheme that moves content files and/or distributes resource accesses based on a predictive workload. The claimed invention, however, is more specific, as claim 19 requires the following subject matter (emphasis supplied):

> "using the values to generate a weighted mapping of web applications to manager processes for the set of CDN servers such that the flit and memory capacities for each CDN server are not exceeded; and
>
> servicing requests at the CDN servers in proportion to the weighted mapping."

For a claim to be obvious, under one analytical approach the subject matter must be fairly characterized as involving a simple substitution of one known element for another or the mere application of a known technique to a piece of prior art ready for the improvement. If the

invention cannot be so characterized, obviousness cannot be made out unless the Office can establish "some articulated reasoning with some rational underpinning" – *viz.*, an apparent reason to combine the known elements in the fashion claimed. See, *KSR Int'l v. Teleflex, Inc.*, 127 S Ct. 1727, 1740-41 (2007). The articulated reasoning can be based on interrelated teachings of multiple patents, market demand, or the background knowledge of one of ordinary skill. Even then, however, obviousness cannot be shown unless the subject matter "as a whole" is found in the alleged combination of references. See, 35 USC 103(a).

Under either analytical approach, the combination of Melchione and Ulrich fails to meet this requirement. A simple substitution of the Ulrich adaptive load balancing scheme into the Melchoine software distributed system might be technically feasible, but the combination would still lack the "weighted mapping" function. Moreover, the Examiner has not shown or suggested that Melchione is an existing system that is otherwise "ready for the improvement" of the adaptive load balancing scheme taught by Ulrich. Further, while the Examiner has provided an articulated reason why the references may be combined, that combination does not include the above-identified limitations.

For this reason, one of ordinary skill in the pertinent art would not find the differences between Melchione/Ulrich "obvious" at the time of the invention here. Thus, the Examiner should withdraw the pending obviousness rejection.

Likewise, and without waiver of any contention regarding the current rejection, dependent claims 21-28 are likewise patentable.

Respectfully submitted,

/David H. Judson/

By: _____

David H. Judson, Reg. No. 30,467

ATTORNEYS FOR APPLICANT

April 16, 2009